PATENT ABSTRACTS OF JAPAN

(11)Publication number : 09-044366
(43)Date of publication of application : 14.02.1997

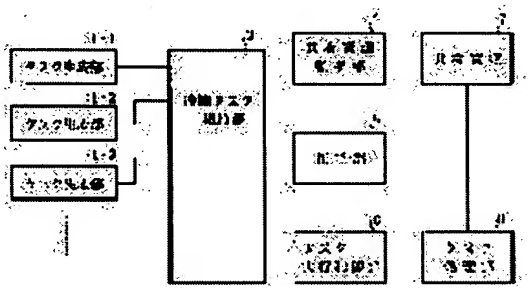
(51)Int.Cl. G06F 9/46
G06F 9/44

(21)Application number : 07-192994
(22)Date of filing : 28.07.1995

(71)Applicant : OKI ELECTRIC IND CO LTD
(72)Inventor : HARADA YOKO

(54) MULTI-THREAD SCHEDULING DEVICE

(57)Abstract:
PROBLEM TO BE SOLVED: To perform proper scheduling of multi-thread in the process processing operated in multithread.
SOLUTION: A thread is assigned to each task generated at an arbitrary timing. Each thread waits for execution in a waiting task holding part 3. An inference part 5 determines the thread to be next executed by fuzzy inference. The priority and the scheduling system are expressed in the range of values indicating their degrees, and fuzzy inference is performed in an optimum range, and the inference result is weighted in accordance with the wait time from generation to execution of the thread to determine the priority of thread execution.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(19)日本国特許庁（ＪＰ）　(12) **公開特許公報（Ａ）**　(11)特許出願公開番号

特開平9－44366

(43)公開日　平成9年(1997)2月14日

(51)Int.Cl.[6]		識別記号	庁内整理番号	ＦＩ		技術表示箇所
Ｇ０６Ｆ	9/46	３４０		Ｇ０６Ｆ	9/46	３４０Ｂ
	9/44	５５４	9189－5Ｂ		9/44	５５４Ａ

審査請求　未請求　請求項の数1　ＯＬ　（全 8 頁）

(21)出願番号　特願平7－192994

(22)出願日　平成7年(1995)7月28日

(71)出願人　000000295
沖電気工業株式会社
東京都港区虎ノ門1丁目7番12号

(72)発明者　原田　洋子
東京都港区虎ノ門1丁目7番12号　沖電気工業株式会社内

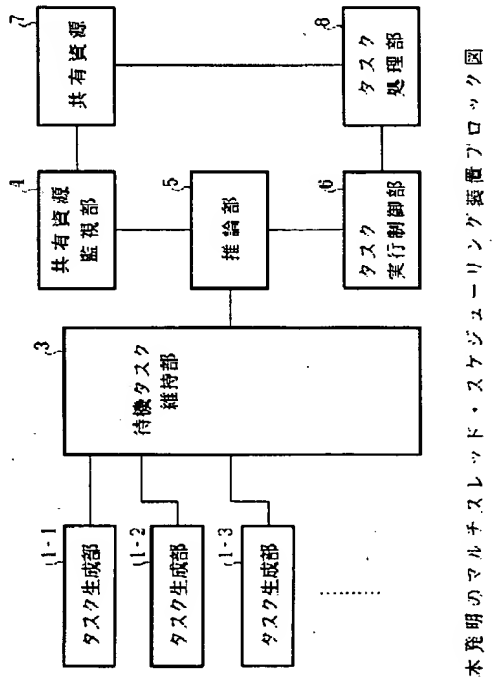
(74)代理人　弁理士　大西　健治

(54)【発明の名称】　マルチスレッド・スケジューリング装置

(57)【要約】

【目的】　マルチスレッドで動作するプロセス処理において、マルチスレッドの適切なスケジューリングを行う。

【構成】　任意のタイミングで生成された各タスクには、それぞれスレッドが割り当てられる。各スレッドは待機タスク維持部３において実行待ち状態となる。推論部５は次に実行すべきスレッドをファジイ推論によって決定する。プライオリティやスケジューリング方式はその度合を表す値の範囲で表現し、最適な範囲でファジイ推論し、スレッドが生成されてから実行されるまでの待機時間から推論された結果に重みを付けてスレッド実行の優先度を決定する。



本発明のマルチスレッド・スケジューリング装置のブロック図

1

【特許請求の範囲】
【請求項１】　任意のプロセスにより生成された複数のタスクに対してそれぞれスレッドが割り当てられ、このスレッドが実行に移されるまでタスクを待機させる待機タスク維持部と、
各タスクが作用を及ぼし合う共有資源と、その共有資源の同期制御変数を監視する共有資源監視部と、
待機中の各タスクの実行の優先順位に関する属性値と待機時間と同期制御変数との関係をルールベースに含めて、各タスクの実行順序をファジイ推論する推論部と、
推論部の推論により得られたタスクの実行順に従って、待機タスク維持部中の各スレッドに実行命令を発するタスク実行制御部と、
このタスク実行制御部より発せられた実行命令に従ってタスクの処理を実行し、共有資源に作用を及ぼすタスク処理部を備えたことを特徴とするマルチスレッド・スケジューリング装置。
【発明の詳細な説明】
【０００１】
【産業上の利用分野】本発明は、複数のタスクをスレッドに割り当てて並列処理を行うマルチスレッド処理において、各スレッドの実行順であるスケジューリングと共有資源保護のための同期を制御するためのマルチスレッド・スケジューリング装置に関する。
【０００２】
【従来の技術】計算機上で生成されるプロセスが複数のタスクをスレッドに割り当てて並列処理を行う場合がある。このような処理をマルチスレッド処理と呼んでおり、高速かつ複雑な演算処理に利用される。このようなマルチスレッド処理において、各スレッドはプロセスによってマッピングされ、一定の資源を共有しながら処理を実行する。従って、これらのスレッドの実行順を調整するスケジューリングはプロセスの結果を正しく得るために重要となる。従来、このようなスケジューリングのために次のような技術が紹介されている（ＯＳＦ　ＤＣＥ技術解説；瓶家喜代志著；株式会社ソフト・リサーチ・センター　第１７９項～２０３項）。ここでは、各スレッドにスケジューリングのプライオリティと、ＦＩＦＯ、ラウンドロビン、スループット、バックグラウンド、アイドルの５種類のスケジューリング方式を組み合わせて、スレッド実行の優先順位を決定し、共有資源保護のため、ミューテックス、コンディション変数の２種類の２値変数を場合によっては複数個ずつ用意して、実行を制御する方法がとられている。
【０００３】
【発明が解決しようとする課題】ところで、上記のような従来のマルチスレッド・スケジューリングの方法には、次のような解決すべき課題があった。上記の方法では、タスク生成時に他のタスクの種類とプライオリティを考慮しながら最も適切なプライオリティを設定しておく必要がある。しかも、不適切なプライオリティを与えるとタスクの実行結果が有効とならないという問題もある。また、タスク自体が実行できない場合も起こり得るという問題もあった。

2

【０００４】ところが、適切なプライオリティを知ることはタスク生成時には容易でなく、またプライオリティの値を多値に設定しても実際に使用されるプライオリティの値は数少なく、プライオリティ値の範囲が有効に使われないという問題があった。また、共有資源保護のためのミューテックス、コンディション変数の使い方によってはタスクがデッドロック状態となり、ある共有資源に対応するタスクが全く処理されない状態に陥ってしまうことがある。これを解決する方法も、またタスク生成時に配慮する必要があり、マルチタスク処理の設計は容易でないという問題があった。
【０００５】
【課題を解決するための手段】本発明は上記の点を解決するため次の構成を採用する。任意のプロセスにより生成された複数のタスクに対してそれぞれスレッドが割り当てられ、このスレッドが実行に移されるまでタスクを待機させる待機タスク維持部と、各タスクが作用を及ぼし合う共有資源と、その共有資源の同期制御変数を監視する共有資源監視部と、待機中の各タスクの実行の優先順位に関する属性値と待機時間と同期制御変数との関係をルールベースに含めて、各タスクの実行順序をファジイ推論する推論部と、推論部の推論により得られたタスクの実行順に従って、待機タスク維持部中の各スレッドに実行命令を発するタスク実行制御部と、このタスク実行制御部より発せられた実行命令に従ってタスクの処理を実行し、共有資源に作用を及ぼすタスク処理部を備える。
【０００６】
【作用】任意のタイミングで生成された各タスクには、それぞれスレッドが割り当てられる。待機タスク維持部において実行待ち状態となる。推論部は次に実行すべきスレッドをファジイ推論によって決定する。プライオリティやスケジューリング方式はその度合を表す値の範囲で表現し、タスクの種類に最適な範囲でファジイ推論し、スレッドが生成されてから実行されるまでの待機時間から推論された結果に重みを付けてスレッド実行の優先度を決定する。
【０００７】
【実施例】以下、本発明を図の実施例を用いて詳細に説明する。図１は、この発明の実施例を示すマルチスレッド・スケジューリング装置のブロック図である。図の装置は、タスク生成部１－１～１－３…と、待機タスク維持部３と、共有資源監視部４と、推論部５と、タスク実行制御部６と、共有資源７及びタスク処理部８から構成される。マルチスレッドで動作することを前提に設計されたプロセスを起動することでタスクが生成される。タ

スク生成部１－１～１－３は、例えばそのプロセスに該当する。生成された各タスクにはスレッドが割り当てられる。待機タスク維持部３は、スレッドが実行に移されるまでタスクを待機させる部分である。

【０００８】共有資源監視部４は、各タスクが共有し作用を及ぼし合う共有資源７とその同期制御変数の状態を監視するとともに、その結果を推論部５に伝える部分である。推論部５は、待機タスク維持部３に維持されているレコードのパラメータと共有資源監視部４から得たパラメータを用いてスレッドの実行順をファジイ推論する部分である。タスク実行制御部６は、共有資源監視部４と、推論部５の処理によるファジイ推論の結果から引き出されたスレッドの実行順に従って待機タスク維持部３のスレッドのタスクに実行命令を発行する部分である。タスク処理部８はタスク実行制御部より発行された実行命令に従ってスレッドの処理を実行し、共有資源７に作用を及ぼす部分である。

【０００９】図２には、待機タスク維持部３の具体的な構成説明図を示す。待機タスク維持部３は、各スレッドに対応するレコードを図のように配列した構成をとる。各レコードはスレッドの属性であるプライオリティ１１と、スケジューリング方式１２、その他の属性１３及び処理１４から構成される。またこの他に、このレコードには、タスク、スレッドが待機タスク維持部３に登録されてからの待機時間１５や実行順位１６が付加されている。なお、このようなレコード３－１～３－３は、タスクが生成される度に追加されていく。

【００１０】図３には、タスク生成部の構成説明図を示す。図に示すように、あるプロセス２０をタスク生成部とすればこのプロセス２０はタスク１０－４，１０－５を任意の数生成する他、タスク１０－５が一定の処理を実行中新たなタスク１０－６を生成する。即ち、１個のプロセスで１つ以上のタスクが生成されることがあり、また、あるタスクの中で更にタスクが生成される場合もある。

【００１１】図４には、推論部５の構成説明図を示す。推論部５は、一定のルールに従って一定の結果を導き出す演算処理装置から構成される。ここにはルール・ベース５－１と、推論エンジン５－２が設けられている。ルール・ベース５－１は、図２に示した待機タスク維持部３の各レコード３－１～３－３のパラメータと、共有資源監視部４から得た同期制御変数等のパラメータを用いてスレッドの実行順位の推論を行うためのルールを規定した部分で、推論エンジン５－２はこのルールに従って実際の推論を行う部分である。

【００１２】図１に示した共有資源監視部４は、共有資源７を保護するためのミューテックス及びコンディション変数を生成し、ミューテックスの値、コンディション変数の値を監視する。タスク実行制御部６は推論部５でスレッドの実行順位が決定されると、これに従って最も

順位の高いスレッドのタスク処理に対し実行命令を出す。また、このスレッドが実行された場合には、このスレッドに関するレコードを待機タスク維持部３から削除する。タスク処理部８はタスク実行制御部６で実行命令が出されたスレッドのタスク処理を実際に実行し、共有資源７に対して作用を及ぼす演算処理部分から構成される。

【００１３】図５は、本発明の装置の動作フローチャートである。このフローチャートを用いて、本発明の装置の動作を順に説明する。マルチスレッド処理が開始されると、並列処理されるべきタスクが順に生成される（ステップＳ１）。これらのタスクは先に説明したように、独自に生成されあるいはネスティングされて任意のタイミングで生成される。同時に生成されるタスクも存在する。１つのタスク生成部から１つのタスクのみが生成される場合もある。

【００１４】生成されたタスクは各々図２を用いて説明したスレッドレコードを生成する（ステップＳ２）。このレコードは、待機タスク維持部３に順に追加されていく。このレコード中のプライオリティ、スケジューリング方式、その他の属性、あるいはタスクは、タスク生成時に設定される。図６には、レコード中のプライオリティの範囲と度合、待機時間の範囲と度合、実行順位の範囲と度合に関する説明図を図示した。

【００１５】図６（ａ）に示すように、プライオリティは、非常に高い（positive large;ＰＬ）、高い（positive medium;ＰＭ）、中程度（approximately zero;ＺＲ）、低い（negative medium;ＮＭ）、非常に低い（negative large;ＮＬ）のメンバシップ関数をとるものとする。スケジューリング方式は、従来方法と同様に先入れ先出し（ＦＩＦＯ）、ラウンドロビン（ＲＲ）、スループット（ＴＲＵ）、バックグラウンド（ＢＧ）、アイドル（ＩＤＬＥ）の５種類とする。これらも上記プライオリティと同様の（ａ）に示すようなメンバシップ関数を持ち、優先順位は非常に高い（positive large;ＰＬ）、優先順位は高い（positive medium;ＰＭ）、優先順位は中程度（approximately zero;ＺＲ）、優先順位は低い（negative medium;ＮＭ）、優先順位は非常に低い（negative large;ＮＬ）に対応させる。なお、スケジューリング方式に関する説明図の図示は省略した。図中、度合というのは、各優先順位のスレッドが発生する確率をいい、プライオリティは、各三角形や台形の底辺の範囲に設定される。

【００１６】一方、待機時間は、スレッドが生成されてからの時間のパラメータであり、レコード生成時には０で初期化され、時間の経過とともに増加する。スレッドが処理を開始したがコンディション変数により待ち状態に入った、あるいは実行順位の高いスレッドが現れた等の理由により、処理が中断した場合、０で初期化される。待機時間は図６（ｂ）に示すように、非常に長い

5

(positive large; ＰＬ)、長い(positive medium;ＰＭ)、やや長い(positive small; ＰＳ)、中程度(approximately zero; ＺＲ)、やや短い(negative small; ＮＳ)、短い(negative medium;ＮＭ)、非常に短い(negative large; ＮＬ)のメンバシップ関数をとる。

【００１７】実行順位は、スレッド生成時に、待機タスク維持部３に登録されているスレッドの中の最下位の順位で初期化する。実行順位は図６(ｃ)に示すように、非常に高い(positive large; ＰＬ)、高い(positive medium;ＰＭ)、中程度(approximately zero; ＺＲ)、低い(negative medium;ＮＭ)、非常に低い(negative large; ＮＬ)のメンバシップ関数をとるものとする。ファジイ推論を行うと、その結果これらの順位が変化することがある。

【００１８】次のステップＳ３では、タスクが生成されたとき、そのタスクが作用を及ぼそうとしている共有資源を保護するためのミューテックス変数の存在を調べる。存在しない場合、共有変数に固有のミューテックス、コンディション変数を、共有資源監視部４内で生成する(ステップＳ４)。従来の方法では、ミューテックス変数はロックされているかいないかの２値変数であったが、本発明の装置ではファジイ推論を行うことで多値化することが可能である。この場合、ミューテックス変数はあるスレッドが共有資源をロックできる優位度を示すものとなる。コンディション変数は共有資源がある状態になるのを待つための機構であり、コンディション変数で待ち状態になったとき、このスレッドは一時的にミューテックスをアンロックする必要があるので、ミューテックス変数によるロックの優位性と連動して変化することになる。

【００１９】ステップＳ５では、優先順位設定の対象となる全てのタスク生成処理が終了しているかを調べ、未終了の場合はステップＳ１のスレッド生成処理から再び処理を再開する。タスク生成処理が終了している場合、ステップＳ１からＳ５の処理は一時停止するが、タスクの生成は非同期的に発生するので、タスク生成要求がある度にステップＳ１から処理が開始される。

【００２０】次に、ステップＳ８のファジイ推論処理は、ステップＳ２で生成されたレコードのうち、プライオリティ、スケジューリング方式、待機時間から、実行順位とミューテックス獲得の優位度を推論する。図１に示す推論部５は、図４に示したようにルール・ベース５－１と推論エンジン５－２で構成される。ルール・ベース５－１に格納されているルールは、例えば次の内容となる。

ルール１： if プライオリティ is ＰＬ andスケジューリング方式 is ＦＩＦＯthen 実行順位 is ＰＬ and ミューテックス is ＰＬ

ルール２： if プライオリティ is ＰＭ andスケジューリング方式 is ＦＩＦＯthen 実行順位 is ＰＭ andミューテックス is ＰＬ

ルール３： if プライオリティ is ＺＲ andスケジューリング方式 is ＦＩＦＯthen 実行順位 is ＺＲ andミューテックス is ＰＬ

・

・

・

ルール２５： if プライオリティ is ＮＬ andスケジューリング方式 is ＩＤＬＥ then 実行順位 is ＮＬ and ミューテックス is ＮＬ

ルール２６： if 待機時間 is ＰＬ then 実行順位 is ＰＬ andミューテックスis ＰＬ

ルール2･7： if 待機時間 is ＰＭ then 実行順位 is ＰＭ andミューテックスis ＰＭ

ルール２８： if 待機時間 is ＰＳ then 実行順位 is ＰＳ andミューテックスis ＰＳ

・

・

・

ルール３２： if 待機時間 is ＮＬ then 実行順位 is ＰＬ andミューテックスis ＮＬ

推論エンジン５－２での推論は、ルール１から２５までから得られる結果とルール２６から３２までから得られる結果に、０から１までの範囲のあるウエイトをかけて、２種類の推論結果に重み付けをし、重心法によって実行順位を決定する。

【００２１】図７と図８に、推論エンジンの具体的な推論方法説明図を示す。ここには、プライオリティがＰＭ、スケジューリングがＰＭ、待機時間がＺＲであった場合の推論の詳細を示した。なお、以下はファジイ推論のための一般的な手法を本発明に適用した例を示す。ルール１は、プライオリティＰＬとこのときに実際に与えられたプライオリティ(fact) ＰＭのand をとり、その結果のα１を次のスケジューリングにマッピングする。ルール１のスケジューリング方式ＰＬとfactＰＭのandをとり、プライオリティから得られたα１とのmin をとる。この値をα２とする。α２を実行順位のＰＬにマッピングし、面積ｗ１を求める。同様にルール２から面積ｗ２、ルール３から面積ｗ３というようにルール２５までの面積を求め、総和Ｓ１を算出する。この面積の総和Ｓ１は、プライオリティとスケジューリング方式とをパラメータとしたときの、このスレッドが優先度ＰＬで実行される可能性に比例する。

【００２２】次に、ルール２６からルール３２より同様に待機時間factとのand をとり、面積の総和Ｓ２を算出する。この面積の総和Ｓ２は、待機時間をパラメータとしたときの、このスレッドが優先度ＺＲとＰＳの重なり部分で実行される可能性に比例する。この実施例ではＳ１とＳ２に０から１の範囲で重みをつけ重み付き総和を

7

算出し、重心法により実行順位を推論する。重要度の低いパラメータに結果が左右され過ぎるのを防止するためである。実行順位と同様にしてミューテックス獲得の優先度をルール１からルール３２より推論し、共有資源監視部４に保存する。

【００２３】図５のステップＳ９の実行順位変更処理では、ステップＳ８のファジイ推論により得られた実行順位を図２のスレッドレコード内の実行順位１６と置き換える。ステップＳ１０のスレッド実行処理では、図２のレコード中の実行順位の最も高いスレッドが実行に移される。実行が開始されたスレッドは必要なミューテックス変数の獲得優先度を同じミューテックスを獲得しようとしている他のスレッドと比較し、ミューテックスの獲得が可能であればこのミューテックスを獲得し、共有資源に対し作用を及ぼす。ミューテックス変数獲得優先度は実行順位よりも重視される。実行されたスレッドの処理はミューテックスの状態により処理が終了しないことも有り得る。このため、ステップＳ１１で処理が完全に終了しているかを調べる。処理が終了した場合、このスレッドのミューテックスに関するパラメータは図２のスレッドレコードとともにステップＳ１２において削除されるが、処理が終了しなかった場合、スレッドレコードの待機時間を０に初期化してファジイ推論の対象スレッドとして残される。ステップＳ１３においては図２のスレッドレコードが０であるかを調べ、処理の終了を判断する。

【００２４】なお、ステップＳ６では処理開始時からの時間経過を常時計測し、スレッドレコードの待機時間を更新するとともに、タスクが新たに生成されなくても一定時間経過した場合はステップＳ８のファジイ推論を行うため、ステップＳ７では一定時間が経過したことを判断し、経過していた場合は、ファジイ推論を開始させる。

【００２５】

【発明の効果】以上説明した本発明のマルチスレッド・スケジューリング装置は、上記の構成により次のような効果が得られる。

(1) タスクを割り当てるスレッドに従来のスレッドに関する情報（属性）の他に待機時間、実行順位を設け、プライオリティ、スケジューリング方式、待機時間で実行順位をファジイ推論するので、最適なパラメータをタスク生成時に付与する必要がなくなり、タスクの設計を容易にすることが可能となった。

8

(2) プライオリティ、スケジューリング方式では、従来の１つの値から度合を表現する多値に範囲を拡張したので、これらの数値の与え方がスケジューリングを大きく左右することがなくなり、タスクの種類を厳密に分類してプライオリティ、スケジューリング方式を決める必要がなくなった。

(3) 待機時間を設け、プライオリティ、スケジューリング方式と同様に度合を表現する多値に拡張すれば、プライオリティ、スケジューリング方式だけで、実行に移されにくいスレッドについても、実行順位が上がるので実行の機会を増やすことが可能になった。

【００２６】(4) 推論結果には重みを付けて最終結論を出すので、待機時間を設けたことで本来意図するプライオリティ、スケジューリング方式に反しても待機時間の長いタスクが処理されることを抑制できる。

(5) ミューテックス獲得を２値から多値に拡張したので、ミューテックス獲得の優先度が下がってくると、スレッドの順位が下がらなくても処理を中断し、他のミューテックスの獲得優先度が大きいスレッドへ処理を移行させることが可能であり、デッドロック状態を回避することが容易になった。

【図面の簡単な説明】

【図１】本発明のマルチスレッド・スケジューリング装置実施例を示すブロック図である。

【図２】待機タスク維持部の構成説明図である。

【図３】タスク生成部の構成説明図である。

【図４】推論部の構成説明図である。

【図５】本発明の装置の動作フローチャートである。

【図６】（ａ）はプライオリティの範囲と度合の説明図、（ｂ）は待機時間の範囲と度合の説明図、（ｃ）は実行順位の範囲と度合の説明図である。

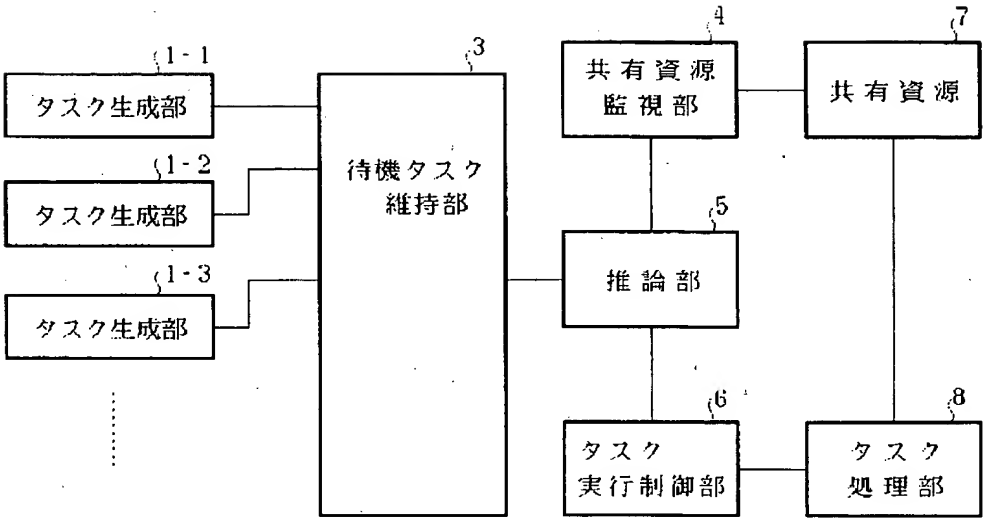
【図７】推論エンジンでの推論方法説明図（その１）である。

【図８】推論エンジンでの推論方法説明図（その２）である。

【符号の説明】

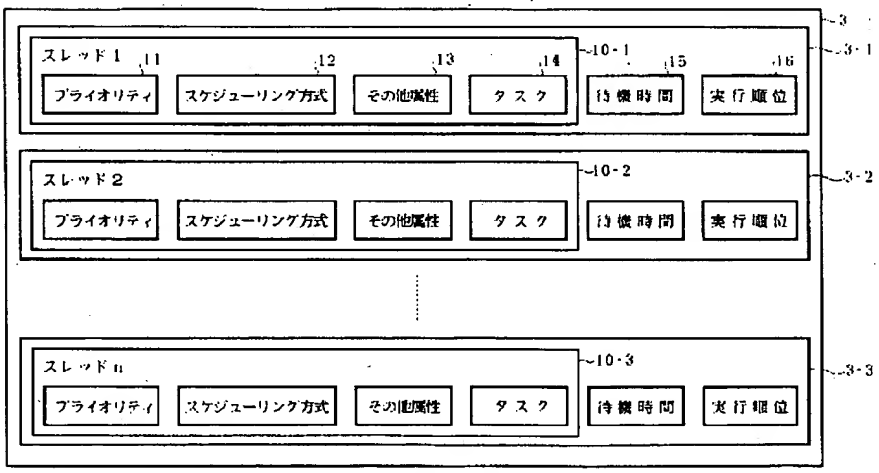
１－１～１－３　タスク生成部
３　待機タスク維持部
４　共有資源監視部
５　推論部
６　タスク実行制御部
７　共有資源
８　タスク処理部

【図１】



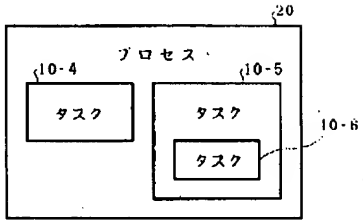
本発明のマルチスレッド・スケジューリング装置ブロック図

【図２】



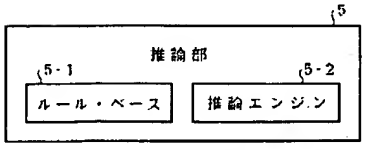
待機タスク維持部の構成説明図

【図３】



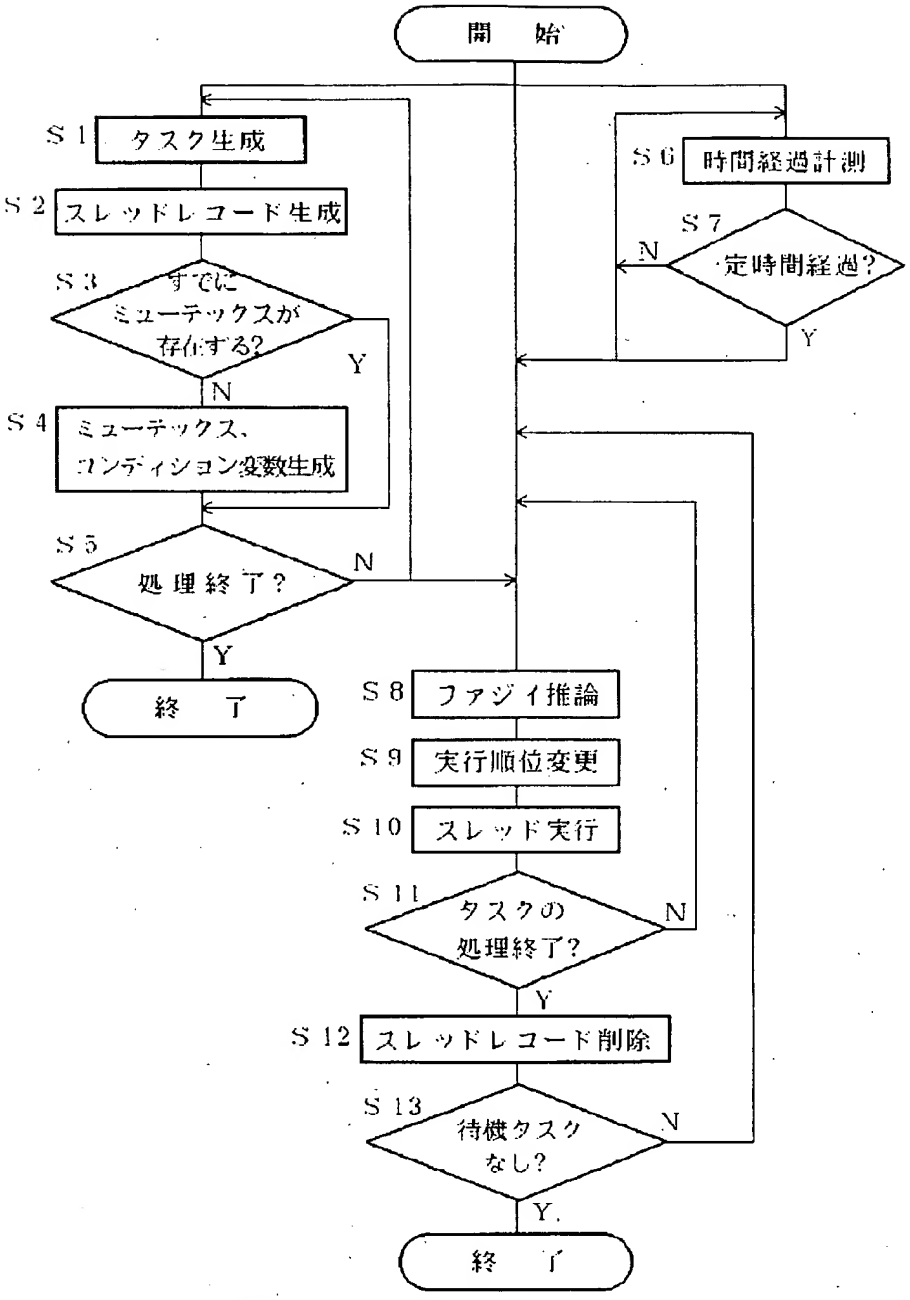
タスク生成部の構成説明図

【図４】



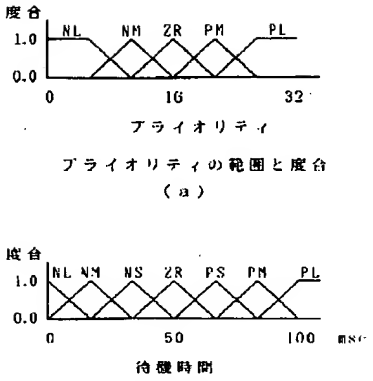
推論部の構成説明図

【図5】



本発明の装置の動作フローチャート

【図６】

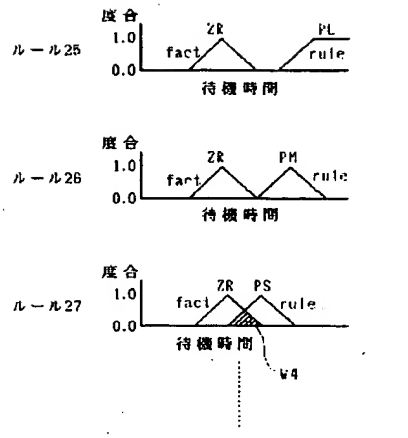


プライオリティの範囲と度合
（ａ）

待機時間の範囲と度合
（ｂ）

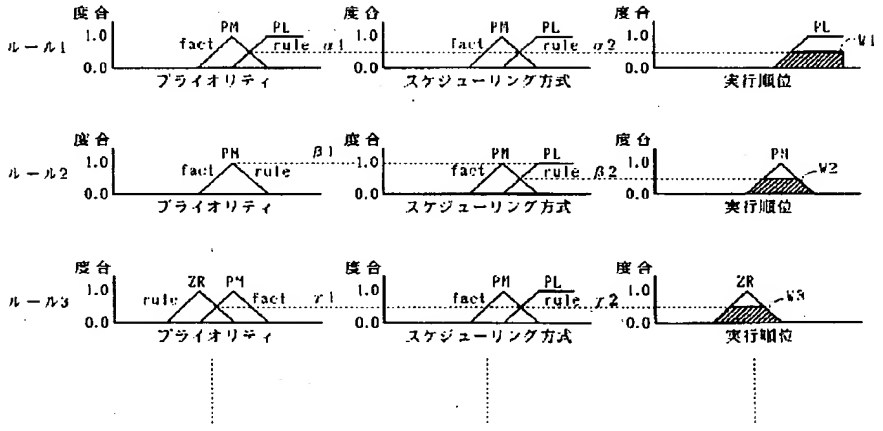
実行順位の範囲と度合
（ｃ）

【図８】



推論エンジンでの推論方法説明図（その２）

【図７】



推論エンジンでの推論方法説明図（その１）